



Figure 10

1100

```
PROCESS (CLK, D, RESET)
BEGIN
```

```
    IF (RESET = '1') THEN
```

```
        Q <= '0';
```

```
    ELSIF (CLK'EVENT AND CLK = '1') THEN
```

```
        Q <= D;
```

```
    END IF;
```

```
END PROCESS
```

Figure 11

1300

```
PROCESS (FAST_CLK)
BEGIN
    IF (FAST_CLK'EVENT AND FAST_CLK = '1')
    THEN
        SAMPLED_CLK <= CLK;
    END IF
END PROCESS;
```

```
CLK_EVENT <= SAMPLED_CLK /= CLK;
CLK_STABLE <= SAMPLED_CLK = CLK;
CLK_LASTVALUE <= SAMPLED_CLK;
```

1310

```
PROCESS (CLK, D, RESET, CLK_EVENT)
    VARIABLE TRACE1, TRACE2 : BIT;
BEGIN
    TRACE1 := '0'; TRACE2 := '0';
    IF (RESET = '1') THEN
        TRACE1 := '1';
        Q <= '0';
    ELSIF (CLK_EVENT AND CLK = '1') THEN
        TRACE2 := '1';
        Q <= D;
    END IF;
    SIG_TRACE1 <= TRACE1; SIG_TRACE2 <= TRACE2;
END PROCESS;
```

1320

```
PROCESS (CLK, D, RESET)
BEGIN
    IF (RESET = '1') THEN
        Q <= '0';
    ELSIF (CLK'EVENT AND CLK = '1') THEN
        Q <= D;
    END IF;
END PROCESS
```

1330

Figure 13

```

1500 ALWAYS @(POSEDGE CLK OR NEGEDGE RESET)
      BEGIN
          IF (RESET == 0)
              Q <= 0;
          ELSE
              Q <= D;
      END

```

Figure 15

```

1600 ALWAYS @(POSEDGE FAST_CLK)
      BEGIN
          SAMPLED_CLK <= CLK
          SAMPLED_RESET <= RESET;
      END

      ASSIGN CLK_EDGE = SAMPLED_CLK ^ CLK;
      ASSIGN RESET_EDGE = SAMPLED_RESET ^ RESET;

      INTEGER TRACE1, TRACE2;
      REG [1:0] SIG_TRACE;
      ALWAYS @(CLK_EDGE OR RESET_EDGE OR CLK OR RESET)
      BEGIN
          TRACE1 = 0; TRACE2 = 0;
          IF((CLK_EDGE == 1) && (CLK == 1) && (RESET_EDGE == 1) && (RESET == 0)) ||
              IF (RESET == 0)
                  TRACE1 = 1;
              ELSE
                  TRACE2 = 1;
          SIG_TRACE[0] = TRACE1;
          SIG_TRACE[1] = TRACE2;
      END

      ALWAYS @(POSEDGE CLK OR NEGEDGE RESET)
      BEGIN
          IF (RESET == 0)
              Q <= 0;
          ELSE
              Q <= D;
      END

```

Figure 16